References

1. K. Scarfone, P. Mell, (2007) Guide to Intrusion Detection and Prevention Systems (IDPS) SP-800-94. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-94.pdf
2. Amazon GuardDuty [Електронний ресурс] – режим доступу ресурсу: https://aws.amazon.com/guardduty
3. Amazon CloudTrail [Електронний ресурс] – режим доступу ресурсу: https://aws.amazon.com/cloudtrail/
4. Amazon GuardDuty Features [Електронний ресурс] – режим доступу ресурсу: https://aws.amazon.com/guardduty/features/

OBJECT-ORIENTED MODEL FOR NEURAL NETWORK DAMAGE DETECTION OF MAIL PACKAGES

Molchanova Maryna
Teacher
momolchanova@gmail.com
Mazurets Oleksandr
Ph.D (Engineering Science), Associate Professor
exe.chong@gmail.com
Klimenko Valeriia
Teacher
ler.klimenko.8@gmail.com
Kuflevsky Evgeny
Postgraduate student
e.kuflevsky2003@gmail.com
Computer Science Department
Khmelnytskyi National University, Ukraine

Over time, the number of mail items increases, which requires effective damage detection methods, as they are more often subject to transport and handling losses. In recent years, significant progress has been observed in the field of computing technologies. This makes it possible to use more complex models of neural networks to solve tasks of automated damage detection with high accuracy. Using neural networks to detect damage allows you to automate this process and ensure a quick response to possible problems [1].

Computer vision covers a variety of tasks, such as segmentation, filtering, classification, scene rendering, object positioning, object detection, video surveillance, and others [2]. The development of computer vision is a key component for the advancement of artificial intelligence and intelligent information technologies.

The main tools on which modern image classification algorithms are based include deep learning and convolutional neural networks (CNN) [3]. Convolutional neural networks are similar to regular neural networks, but designed for image processing.

The purpose of the work is to develop an applied object-oriented implementation of the method of detecting damage to the packaging of postal items using the ResNet50 neural network.

The ResNet architecture is considered one of the most popular convolutional neural network architectures. The architecture of the neural network is shown in Figure 1. Deeper neural networks produce higher training and testing error than their shallower counterparts due to the degradation problem. To solve this problem, residual networks use identity mapping to the input layer of the reference layer. Then ResNet can gain accuracy by increasing depth and easy optimization [4].

The problem is commonly known as the vanishing/exploding gradient problem. ResNet, thanks to its architecture, does not allow these problems to occur at all. Pass connections (below) do not allow this, as they act as gradient superhighways, allowing it to flow without significant change.

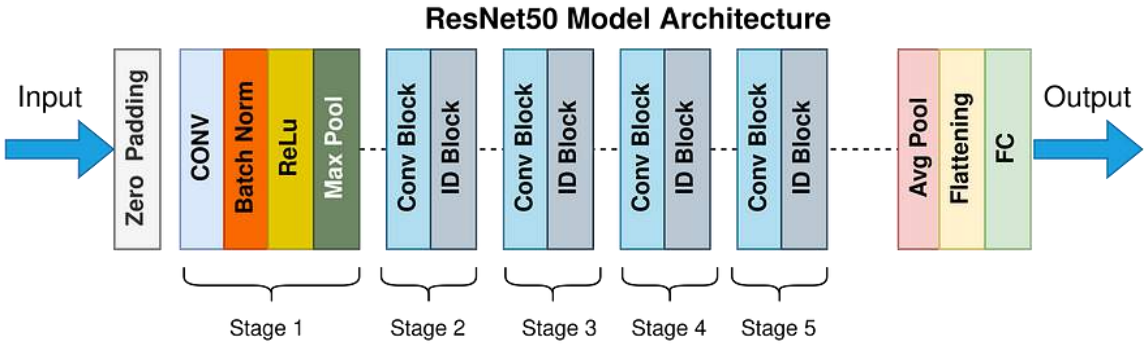

Figure 1. Architecture of ResNet50 neural network [4].

Based on the material reviewed, the CNN neural network, namely its ResNet version, which is resistant to the decaying gradient problem, will be used to identify mail package damage.

The mail package damage detection information system implements mail package damage detection using a ResNet50 neural network, and transforms the input data in the form of a trained CNN model and a test image for classification into output data in the form of a numerical estimate of the probability of the presence of package damage. It consists of three subsystems and a set of photographic images. The diagram of information system classes is shown in Figure 2.

To implement the "Subsystem for working with multiple photographic images", the "ImageGalery" class was designed, which implements the following basic functions:

– selection of a directory for viewing available graphic images ("load_image()" method);

– displaying information about the current image ("load_image_info()" method);

– adding a new image to the set of images ("on_add_images()" method);

– delete the selected image ("on_delete_images()" method).

To implement the "Neural Network Learning Subsystem", a corresponding "ModelTrainer" class was designed, which is designed to perform the following functions:

– neural network training ("build_model()" and "train_model()" method);

– saving the trained neural network model ("save_model()" method);

– deriving an estimate of the performance of the trained model ("evaluate_model()" method).

For the implementation of the "Subsystem for the detection of damage to the packaging of postal items", the appropriate class "DamageDetectionApp" has been designed, which is designed to perform the following functions:

– loading an image for analysis and its output for viewing (method on_choise_image() and display_selected_image());

– neural network assessment of packaging damage (execute_analysis() method).

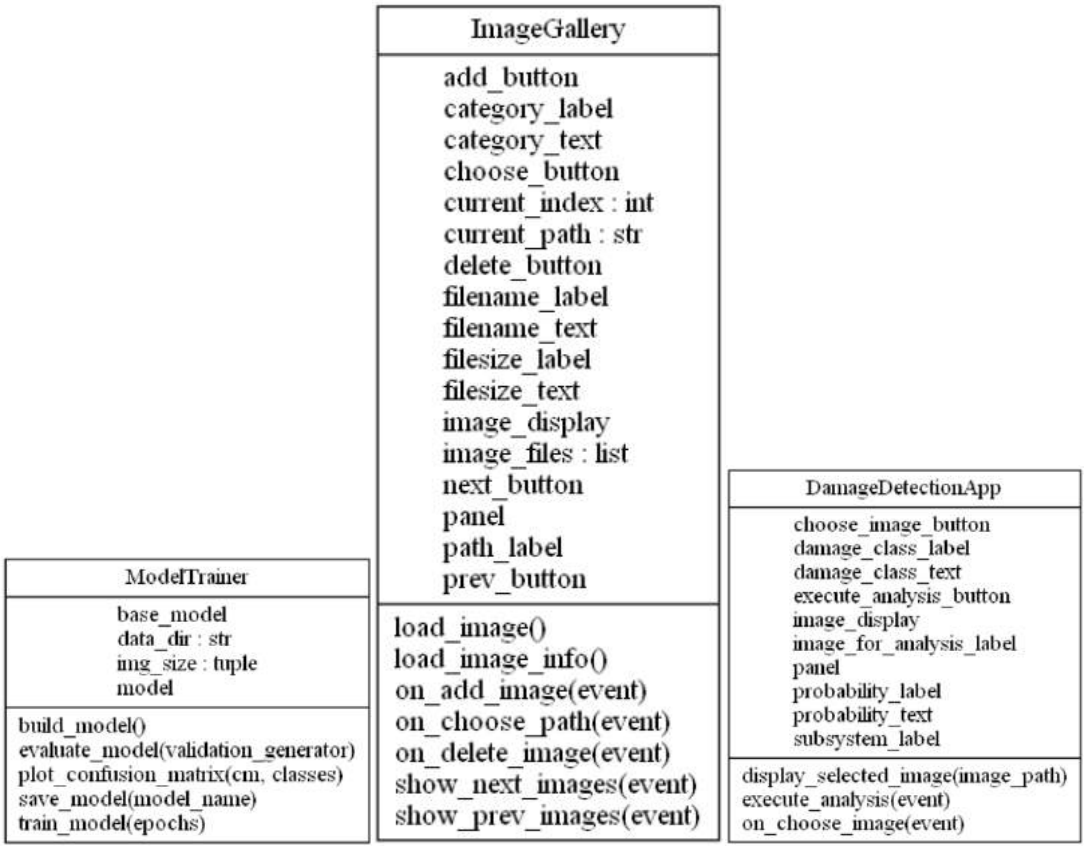

Figure 2. Object-oriented model class diagram for damage detection of mail packages

On the basis of the designed software architecture of the information system for detecting damage to postal packages, which is an implementation of the proposed method of the same name for detecting damage to postal packages, the component parts are implemented.

For correct operation, the "Neural network training subsystem" is first implemented based on the appropriate "ModelTrainer" class. This subsystem does not have a graphical user interface and is an auxiliary, but key component. The result of her work is a preserved trained neural network, which will serve as input data for the main subsystem for detecting damage to the packaging of postal items.

The build_model() method is designed to build and compile a deep neural network (CNN) to solve the problem of detecting mail package damage using the provided basic neural network model (ResNet50). First, an object of the Sequential

model is created, which is a sequence of layers in a neural network, which includes layers in turn. The first layer is to add the basic ResNet50 model. The next layer adds a global mean pooling layer, which reduces the spatial dimensions of the input data by averaging over all positions. After the pooling layer, a fully connected layer with 512 neurons and the ReLU activation function is added to the model. Similarly, an output fully connected layer with one neuron using the sigmoid activation function is added. Since this is a binary classification (damage detected or not detected), sigmoid is used to obtain the probability of exiting the damage presence class. A model is compiled using the Adam optimizer, the binary_crossentropy loss function, and an accuracy metric to evaluate model performance.

After starting this component, the process of neural network training will be displayed in the console (Figure 3).

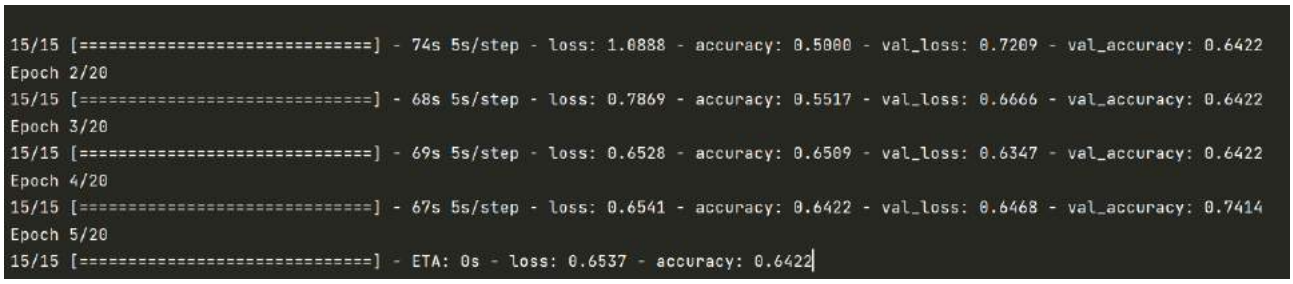

Figure 3. CNN training process.

The train_model() method provides the entire model training process using data generators for training and validation sets, as well as using image augmentation to improve model robustness. The implemented method has a number of features. As part of the method, ImageDataGenerator objects are created, which are used to augment images in the training and validation sets, respectively. Augmentation is the process of creating new images by applying various transformations to the original images, which helps to improve the overall capability of the model to different inputs.

Generators are also created for training and validation data sets, respectively. Generators are iterators that supply batches of data to the model during training. They use ImageDataGenerator to automatically load and augment images from their respective data directories.

The fit() method is used to train the model. The model is trained using the training data generator (train_generator) during the specified number of epochs (as part of the work, the number of epochs varied from 10 to 30).

The evaluate_model() method is used to evaluate the performance of the constructed model on the validation data, outputting a classification report, a confusion matrix, and a visualization of this matrix.

The subsystem responsible for detecting packaging damage is based on the use of a neural network that was trained on the basis of the neural network learning subsystem. This subsystem already has a graphical user interface, which is written using the "WX" library.

The on_choose_image() method opens a file selection dialog for selecting an image, gets the path of the selected file, and calls a method to display the selected image.

The execute_analysis() method is designed to perform an analysis of the selected image and set the appropriate values for the damage class and probability. The interface of the software component is shown in Figure 4.

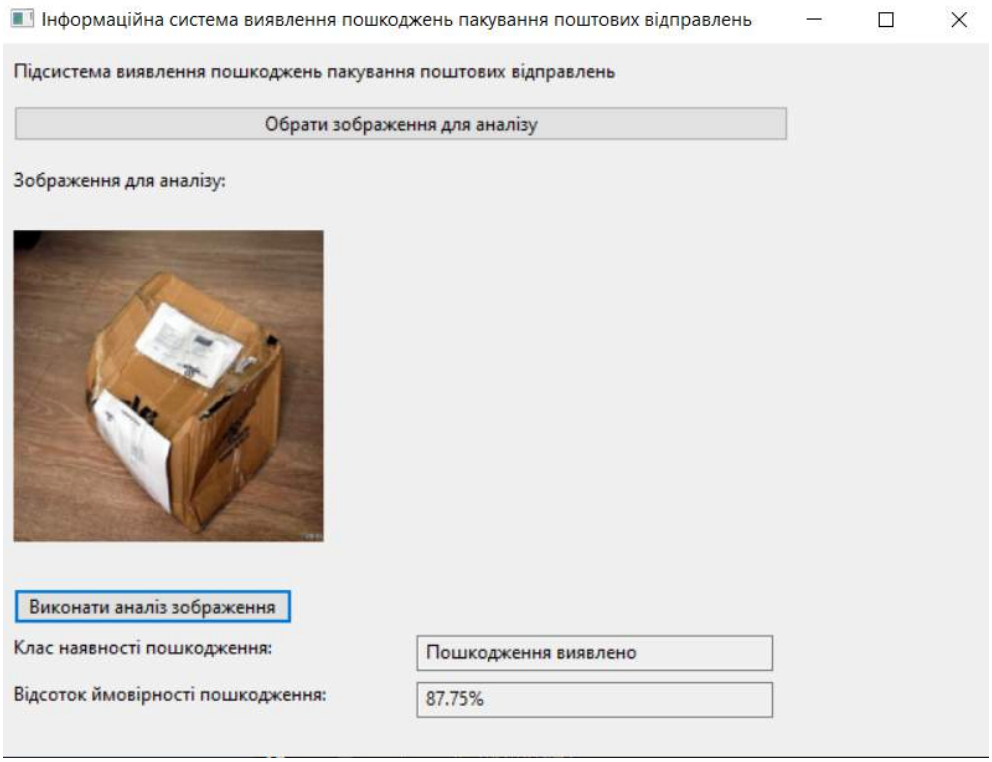

Figure 4. The interface of object-oriented subsystem for detecting damage to the packaging.

So, in this way, the software structure of the object-oriented information system was designed and the functional purpose of the software components of the mail package damage detection system, which consists of three main classes "DamageDetectionApp", "ModelTrainer", "ImageGalery", was described. The components of the object-oriented information system for detecting damage to postal packages, consisting of three classes, which are responsible for the implementation of subsystems, were implemented.

References

1. Defect Detection in Packaging: Computer Vision to the Rescue. URL: https://blog.gramener.com/defect-detection-in-packaging/
2. O. Barmak, O. Mazurets, I. Krak, A. Kulias, Method for automated test tasks creation for educational materials. CEUR Workshop Proceedings, 2020, vol. 2711, pp. 309–320. URL: http://ceur-ws.org/Vol-2711/paper24.pdf
3. Convolutional neural networks, explained. URL: https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939
4. ResNet-50 URL: https://medium.com/@arashserej/resnet-50-83b3ff33be7d